AF/2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

OIPE MAR 19 2004 PATENT & TRADEMARK OFFICE

In re application of: **Ault, Jr.** §
Serial No: **09/820,462** §
Filed: **March 29, 2001** §
For: **Generating Partials for Perspective Corrected Texture Coordinates in a Four Pixel Texture Pipeline** §

Group Art Unit: **2676**
Examiner: **Tran, Tam D.**
Attorney Docket No.: **AUS920000612US1**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on 3/16/04.
By: Kit Cruz-Verona
Kit Cruz-Verona

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

TRANSMITTAL DOCUMENT

RECEIVED
MAR 23 2004
Technology Center 2600

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of $330.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

Duke W. Yee
Registration No. 34,285
CARSTENS, YEE & CAHOON, LLP
P.O. Box 802334
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANT

4/Brief 3/24/01 i983

Docket No. AUS920000612US1 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Ault, Jr.**	§	Group Art Unit: **2676**
	§	
Serial No. **09/820,462**	§	Examiner: **Tran, Tam D.**
	§	
Filed: **March 29, 2001**	§	
	§	
For: **Generating Partials for Perspective Corrected Texture Coordinates in a Four Pixel Texture Pipeline**	§ § § §	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on 3/16/04.

By: Kit Cruz-Verona
Kit Cruz-Verona

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 17, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

03/22/2004 AWONDAF1 00000085 090447 09820462
01 FC:1402 330.00 DA

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party:

International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-15

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-15
4. Claims allowed: NONE
5. Claims rejected: 1-15

C. CLAIMS ON APPEAL

The claims on appeal are: 1-15

STATUS OF AMENDMENTS

All of the amendments to the claims have been entered. No after final amendments were made in this case.

SUMMARY OF INVENTION

The present invention provides a method, program and apparatus for generating partial differential equations for perspective corrected texture coordinates in a computer graphics display. (*Specification*, page 3, lines 4-7). The present invention comprises calculating texture coordinates for four adjacent pixels and then determining the differences between the coordinates. (*Specification*, page 3, lines 7-10). A perspective correction factor is then calculated, which is multiplied by each coordinate difference. (*Specification*, page 3, lines 10-12).

ISSUES

The issues on appeal are:

1. Whether claims 1-15 are properly rejected under 35 U.S.C. § 102(e) as being anticipated by *Piazza et al.* (U.S. Patent No. 6,204,857 B1).

GROUPING OF CLAIMS

The claims do not stand or fall together as a single group. The claims stand or fall based on the following grouping of claims:

Group A: claims 1-3, 5-8, 10-13, and 15

Group B: claims 4, 9, and 14

ARGUMENT

I. Rejection of claims 1-15 under 35 U.S.C. § 102(e), Anticipation

The Examiner has rejected claims 1-15 under 35 U.S.C. § 102(e) as being anticipated by *Piazza et al* (US Patent No. 6,204,857 B1). This rejection is respectfully traversed.

A prior art reference anticipates the claimed invention under 35 U.S.C. §102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.,* 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

Group A: Claims 1-3, 5-8, 10-13, and 15

The *Piazza* reference cited by the Examiner does not anticipate the present invention as recited in claims 1-3, 5-8, 10-13, and 15 because *Piazza* fails to teach each and every element of the claims.

With regard to independent claims 1, 6, and 11, the Examiner states:

> In regards to claim 1, 6, 11, *Piazza* teaches a method and apparatus for generating partial differential equations for perspective corrected texture coordinates, see col. 6, lines 54-62, comprising: calculating texture coordinates at each of four adjacent pixels; see col. 6, lines 60-67, calculating the difference between the texture coordinates; calculating a perspective correction factor based on perspective correction coordinates; and multiplying each texture coordinate difference by the perspective correction factor, see col. 7, lines 1-30.

(*Final Office Action*, dated September 17, 2003, page 2). Independent claim 1, which is a representative claim, reads as follows:

> 1. A method for generating partial differential equations for perspective corrected texture coordinates, comprising the steps of:
> a) calculating texture coordinates at each of four adjacent pixels;
> b) calculating the difference between the texture coordinates;
> c) calculating a perspective correction factor based on perspective correction coordinates; and

d) multiplying each texture coordinate difference by the perspective correction factor;

wherein steps a) through d) include sharing data from each of the four adjacent pixels.

Claim 1 of the present invention recites a series of calculating and multiplying steps that include sharing data from four adjacent pixels. Claim 1 teaches using the shared data in calculating texture coordinate values at the four adjacent pixels to generate partial differential equations for perspective corrected texture coordinates. The values generated are mathematically exact values, because of data that is common among adjacent pixels.

Webster's Ninth New Collegiate Dictionary defines adjacent as "not distant: nearby", "having a common endpoint or border", and "immediately preceding or following". Webster's Ninth New Collegiate Dictionary further clarifies that adjacent "may or may not imply contact but always implies an absence of anything of the same kind in between". Thus, it is apparent that the present invention provides a process for using shared data in calculating exact texture coordinate values at four pixels which, in view of the definition above, do not have any other pixels placed between those four pixels.

In contrast, *Piazza* teaches calculating texture coordinates using four corner pixels, with additional pixels placed in between the four corner pixels. As shown in the passage below, *Piazza* teaches using four corner pixels to calculate texture coordinates through interpolation:

> Particularly, in the method of the invention, a span of a predetermined number of pixels is defined in pixel (screen space). As described, the designators x and y represent the row (x) and column (y) designation of a pixel (picture element) from a plurality or matrix of pixels of a screen or display device. The pixels of the display device are typically rectangular or square and arranged in horizontal rows and vertical columns for convenience. In the preferred embodiment, the predefined span includes a grid of 4.times.4 pixels. Then, a perspectively correct determination of the texture value (u,v) is made at each of four corners of the span, three of the corners being immediately outside the span. As used herein, the designators u and v represent the row (u) and column (v) designation of a texel (texture element or cell) from a plurality of texels of a texture map at a particular LOD. Texels may also be square or rectangular and envisioned to be located in horizontal rows and vertical columns. Then, a **linear interpolation technique** is performed to approximate the texture values and partial derivatives of texture addresses of each pixel within the span that map to the polygon to be rendered.

(*Piazza*, col. 3, line 66 to col. 4, line 19 (emphasis added)). As can be seen above, the method of generating texture address partial derivatives in *Piazza* includes using interpolation. Interpolation is defined in the Free On-line Dictionary of Computing as:

> A mathematical procedure which estimates values of a function at positions between listed or given values. Interpolation works by fitting a "curve" (i.e. a function) to two or more given points and then applying this function to the required input.

(http://foldoc.doc.ic.ac.uk/foldoc/index.html). The word *interpolate* means to estimate intermediate values occurring between known values. Thus, performing an interpolation technique results in estimated values rather than exact values. As shown above, *Piazza* teaches defining a span (collection of pixels on a single scan line which lie inside the primitive) and performing a linear interpolation technique using the four corners of the span as the known values. *Piazza* teaches using interpolation to "approximate the texture values and partial derivatives of texture addresses of each pixel within the span." Thus, *Piazza* differs from the present invention as recited in claim 1 by performing interpolation to generate approximate or estimated texture values, instead of using shared data to generate exact texture values.

Furthermore, one of ordinary skill in the art would not be motivated to make the changes in *Piazza* to reach the presently claimed invention. The cited sections of *Piazza* provide no teaching, suggestion, or incentive for using shared data from each of the four adjacent pixels to generate mathematically exact texture values. *Piazza* actually teaches away from the presently claimed invention because it teaches using interpolation to achieve approximate values for the texture coordinates in the pixels as opposed to using four adjacent pixels to generate mathematically exact texture values as in the presently claimed invention. Thus, *Piazza* does not teach or suggest the feature of using shared data from each of the four adjacent pixels to generate mathematically exact texture values as recited in claim 1 of the present invention.

In sum, the present invention as recited in claim 1 teaches using shared data from each of the four adjacent pixels to generate texture values in order to reduce the number of multiplies and subtracts used to calculate the texture coordinate values. The values generated from the steps recited in claim 1 are mathematically exact values, because of data that is common among adjacent pixels. In contrast, *Piazza* employs an interpolation technique which generates estimated texture values, or simply offers an average of the pixels placed between corner pixels

having known values. Generating average texture coordinate values over a span is not the same as generating exact texture coordinate values using data shared from four adjacent pixels.

Therefore, Applicants submit that *Piazza* does not teach all elements of rejected independent claims 1, 6, and 11. Furthermore, dependent claims 2, 3, and 5 depend from independent claim 1. Dependent claims 7, 8, and 10 depend from independent claim 6. Dependent claims 12, 13 and 15 depend from independent claim 11. Applicants have already demonstrated claims 1, 6, and 11 to be in condition for allowance. Applicants respectfully submit that claims 2-3, 5, 7-8, 10, 12-13, and 15 are also allowable, at least by virtue of their dependency on allowable claims. Accordingly, Applicants respectfully submit that claims 1-3, 5-8, 10-13, and 15 are patentable over the *Piazza* reference.

Group B: Claims 4, 9, and 14

Piazza does not anticipate the present invention as recited in claims 4, 9, and 14 because *Piazza* fails to teach each and every element of the claims. The arguments made with respect to Group A claims apply to the claim in Group B as well. As mentioned previously, *Piazza* fails to teach the present invention's features of using shared data from each of the four adjacent pixels to generate mathematically exact texture values. These claims include other patentable features in addition to the claims in Group A.

Claim 4 of the present invention recites the additional feature of requiring 12 subtracts and 32 multiplies when calculating the partial differential equations for a single texture coordinate for all four pixels. Claim 4, which is a representative claim, reads as follows:

> 4. The method according to claim 1, wherein calculating the partial differential equations for a single texture coordinate for all four pixels requires 12 subtracts and 32 multiplies.

In the rejection of claim 4, the Examiner refers to the following portion of the *Piazza* reference:

> If, at step 216, it is determined that the first pixel of the first row is not to be texture mapped, then the process continues at step 220 where a determination is made as to whether the second pixel of the first row is lit. If the second pixel of the row is lit, then an interpolation is made at step 222 to determine the texture address partial with respect to the y dimension, i.e., DC1, for the second pixel of the first row of the span, calculated as DC1=DC0+((DR-DL)/4). If DR0 was not calculated in a previous step, then the DR0 calculation DR0=((R0-L0)/4) is

> performed, with DR, DL, R0 and L0 already computed. Additionally, at step 222, the (u,v) texture address value is calculated, which for exemplary purposes, for the texture coordinate U.sub.0,1 is calculated as U01=L0+((R0-L0)/4). After calculating DC1, DR0, and U01 (V01) for the second pixel of the first row, the algorithm proceeds to step 225, FIG. 3(b).

Piazza, col. 8, lines 5-20. In the cited portion, *Piazza* teaches calculating the texture address partial with respect to the *y* dimension for each pixel in the first row of the span. However, this teaching is not equivalent to requiring 12 subtracts and 32 multiplies when calculating the partial differential equations for a single texture coordinate for all four adjacent pixels, as recited in claim 4. The Examiner proffers no analysis as to why simply calculating the texture address partial with respect to the *y* dimension for each pixel in the first row of the span is equivalent to requiring 12 subtracts and 32 multiplies when calculating the partial differential equations for a single texture coordinate for all four adjacent pixels.

In view of the above, applicant submits that claims 4, 9, and 14 also are not anticipated by the *Barbara* reference.

CONCLUSION

In view of the comments above, it is respectfully urged that the rejection of claims 1-15 not be sustained.

Cathrine K. Kinslow
Reg. No. 51,886
Carstens, Yee & Cahoon, LLP
PO Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicant

APPENDIX OF CLAIMS

The claims involved in the appeal are:

1. (previously presented) A method for generating partial differential equations for perspective corrected texture coordinates, comprising the steps of:

a) calculating texture coordinates at each of four adjacent pixels;

b) calculating the difference between the texture coordinates;

c) calculating a perspective correction factor based on perspective correction coordinates; and

d) multiplying each texture coordinate difference by the perspective correction factor;

wherein steps a) through d) include sharing data from each of the four adjacent pixels.

2. (original) The method according to claim 1, wherein the step of calculating the difference between coordinates uses a set of subtracts in parallel.

3. (original) The method according to claim 1, wherein the step of calculating the perspective correction factor uses a set of multiplies in parallel.

4. (original) The method according to claim 1, wherein calculating the partial differential equations for a single texture coordinate for all four pixels requires 12 subtracts and 32 multiplies.

5. (original) The method according to claim 1, wherein subsequent coordinates are found by adding the partial differential equations in a given direction.

6. (previously presented) A computer program product in a computer readable medium for use in a data processing system, for generating partial differential equations for perspective corrected texture coordinates, the computer program product comprising:

a) first instructions for calculating texture coordinates at each of four adjacent pixels;

b) second instructions for calculating the difference between the texture coordinates;

c) third instructions for calculating a perspective correction factor based on perspective correction coordinates; and

d) fourth instructions for multiplying each texture coordinate difference by the perspective correction factor;

wherein instructions a) through d) further include instructions for sharing data from each of the four adjacent pixels.

7. (original) The computer program product according to claim 6, wherein the instructions for calculating the difference between coordinates use a set of subtracts in parallel.

8. (original) The computer program product according to claim 6, wherein the instructions for calculating the perspective correction factor use a set of multiplies in parallel.

9. (original) The computer program product according to claim 6, wherein calculating the partial differential equations for a single texture coordinate for all four pixels require 12 subtracts and 32 multiplies.

10. (original) The computer program product according to claim 6, wherein subsequent coordinates are found by adding the partial differential equations in a given direction.

11. (previously presented) A system for generating partial differential equations for perspective corrected texture coordinates, comprising:

a) a first calculating component which calculates texture coordinates at each of four adjacent pixels;

b) a second calculating component which calculates the difference between the texture coordinates;

c) a third calculating component which calculates a perspective correction factor based on perspective correction coordinates; and

d) a multiplying component for multiplying each texture coordinate difference by the perspective correction factor;

wherein components a) through d) share data from each of the four adjacent pixels.

12. (original) The system according to claim 11, wherein the second calculating component which calculates the difference between coordinates uses a set of subtracts in parallel.

13. (original) The system according to claim 11, wherein the third calculating component which calculates the perspective correction factor uses a set of multiplies in parallel.

14. (original) The system according to claim 11, wherein calculating the partial differential equations for a single texture coordinate for all four pixels requires 12 subtracts and 32 multiplies.

15. (original) The system according to claim 11, wherein subsequent coordinates are found by adding the partial differential equations in a given direction.